c. Methods

i. Subject/Participants

The population in this dataset is composed of demographic information of customers or applicants for house loans.

ii. Data Collection Approaches/Strategies/Data Structures

All the information collected came from the answered online form of Alpha Financing Company, and downloaded in a .csv file. It includes 2 sheets: named Training and Testing. Their structure is the same except the Testing sheet does not have the Loan_Status variable.

To initially investigate the data structure, the file is converted into excel format, and imported in R, using excel_sheets() function. To choose the specific sheet, use the read_excel() function. To return the column headers, use colnames() function, and str() for structure.

Code:
```
#Load the data
excel_sheets("INSURANCE.xlsx")
hloan <-read_excel("INSURANCE.xlsx", sheet="TRAINING")
colnames(hloan)
str(hloan)
```

This returns the demographic information collected from the applicants in numeric and character types.

Table 1 Dataset Structure

Information/Column Header	Data Type	Scale
Applicant Income, Loan Amount, Coapplicant Income	Numeric	thousands (x1000)
Loan_Amount_Term	Numeric	Months
Dependents	Numeric	Unit
Loan_ID, Married, Gender, Education, Self_Employed,Property_Area, Loan_Status	Character	

Structure summary:

- Dimension: 614 rows, 13 columns
- Class: Dataframe

Independent Variable/Predictors/Variable Types:

Categorical Factors (5):	Gender, Married, Dependents, Self_Employed, Credit_History
Ordinal Factors(2):	Education, Property_Area
Continuous (4):	ApplicantIncome, CoapplicantIncome, Loan_Amount_Term, Dependents

Dependent Variable: Loan_Status
Type of Response: Categorical/Classification (Y/N);
can be binomial if converted to 1/0.

iii. Data Analysis Approaches and/or Software

Since Machine learning algorithms learn from data, it is highly important that the data is highly relevant for the objectives we want to meet. This also means that raw data may need to be preprocessed: selected, cleaned, formatted, and scaled or transformed with the purpose of boosting the accuracy of the model.

Here, we discuss the summary of data preparation steps that need to be undertaken relating to visual investigation of data structure, data wrangling, data visualization, and statistical analysis.

Also, to give a general understanding of methodology in approaching the business problem, a flowchart below is created.

Figure 1 Data Analytics and Modeling Methodology

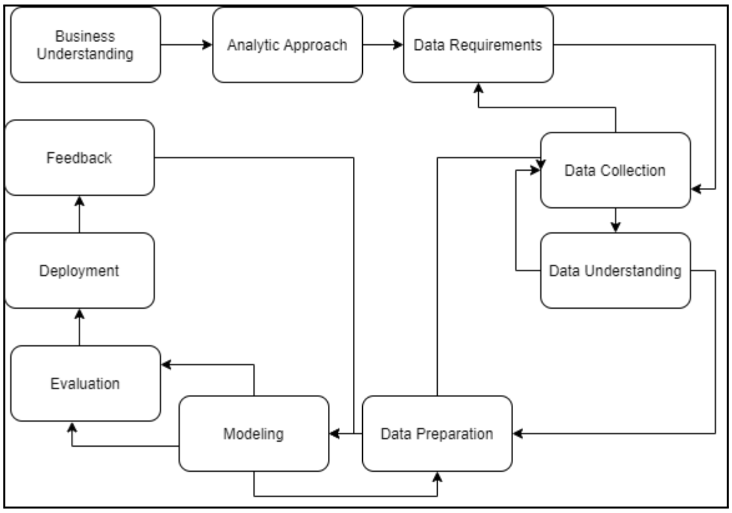
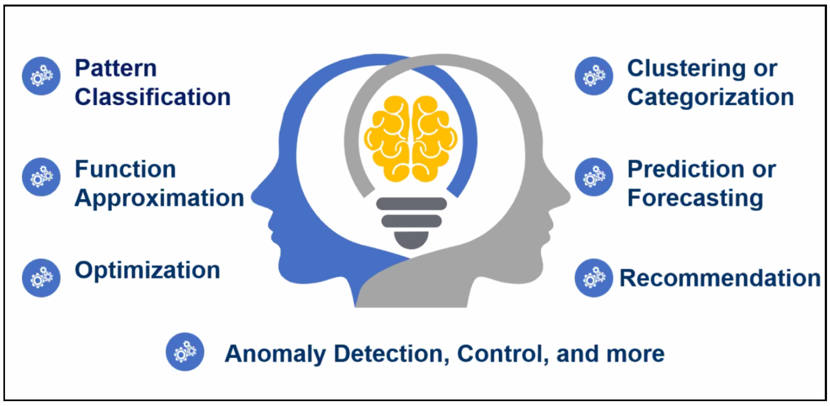


Figure 2 Problems to Solve Using Machine Learning



[1]

iii.a Exploratory Data Analysis (R, Power BI)

This involves visual investigation of file format, overall data structure, data quality, and relationships of involved variables. As we already know that the file is in csv format, the softwares compatible for this step may be as simple as MS Excel, or Power BI. Initial data processing may involve :

- **Formatting** - reformatting data according to the software compatibility
- **Data selection**- to choose data based on the relevance of data to the objectives. It may also involve adding relevant data that is not on the database
- **Data Wrangling**
  - Cleaning -removal or imputation of missing data

[1] A slide taken from Dr Robert G De Luna's webinar Understanding AI: Machine Learning and Deep Learning

- Redesign the data into a usable and functional format and correct/remove any bad data.

  Investigate accuracy of values, and formats. Once the accuracy of values, and format are determined, then they can be transformed into a format that will make analysis or visualization easier

  Using R, variables in character format are transformed into factors:

Code:
```
#Convert character types to factors
cols <- c(1,2,3,5,6,11,12,13)
loan[cols] <- lapply(hloan[cols], factor)
```

- **Data sampling** - will be done if the number of total observations is more than enough for machine learning or statistical analysis.

  Since we only have 614 observations, that is considered enough for analysis.

iii.a.1 Investigation and Treatment of nulls and outliers

- Investigate dataset for existence of null data and possible outliers.

  Using sapply() function, we can have a summary of number of null values per column

Code:
```
#Determine how many missing (na's) are there per column
sapply(hloan, function(x) sum(is.na(x)))
```

```
> sapply(hloan, function(x) sum(is.na(x)))
          Loan_ID            Gender           Married        Dependents         Education 
                0                13                 3                15                 0 
    Self_Employed   ApplicantIncome CoapplicantIncome        LoanAmount  Loan_Amount_Term 
               32                 0                 0                22                14 
   Credit_History     Property_Area       Loan_Status 
               50                 0                 0 
> 
```

Visualization packages or tools for missing data may be used such as naniar package in R.

- Nulls may be treated by imputation using MICE package in R, omitted, or imputed by means, median or mode.
- Outliers may be removed, or the dataset will be log,or scale transformed.
- Reasons for nulls and outliers deletion may also be assumed. A decision to keep, transform or delete will be stated accordingly.

Visualizing Missing Values:

It is important to visualize and determine the summary and relationship of missing values if there's any before deciding to eliminate rows or columns.

Code:

```
#Load library for missing variables
library(naniar)
library(VIM)
#Visualize missing variables
vis_miss(hloan, cluster = TRUE)
vis_miss(hloan, sort_miss = TRUE)
n_var_miss(hloan)
gg_miss_upset(hloan)
gg_miss_upset(hloan,
              nsets = 10,
              nintersects =40)
gg_miss_var(hloan, show_pct = TRUE)
```

Table 3 Missing Values Visualization

Figure 3 Data Map of Missing Values

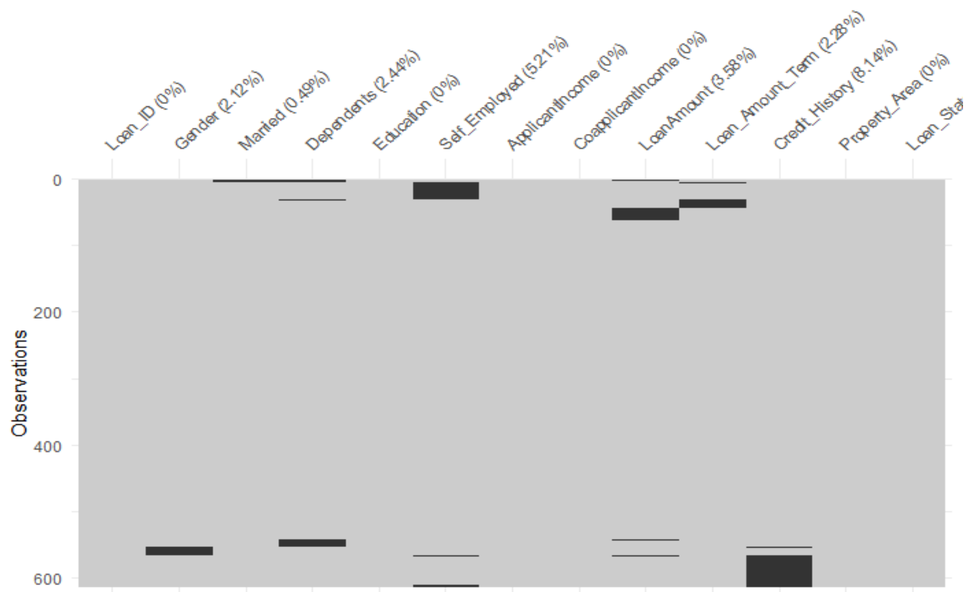
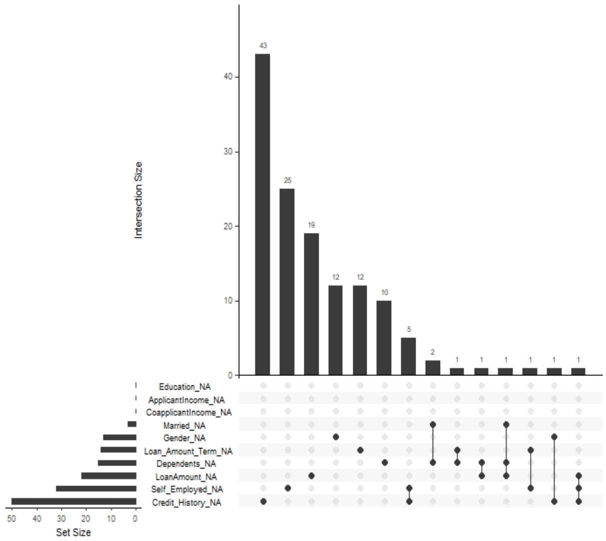


Figure 4 Intersection of null values



Utilizing R's naniar library, we know that:

- 7 variables have missing values.
- Total missing data is at 1.9% of the total.
- Credit history has the most numbers of null values.
- No column has more than 50% missing variables, so in this case, every variable is considered significant for analysis.

iii.a.1.1 Treatment of Null values:

There are 2 ways of filling nulls and both ways may be used to compare effectiveness in creating accurate predictive models.

- Use Multivariate Imputations by Chained Equations (MICE) package in R
  The mice package implements a method to deal with missing data. The package creates multiple imputations (replacement values) for multivariate missing data. The method is based on Fully Conditional Specification, where each incomplete variable is imputed by a separate model. The MICE algorithm can impute mixes of continuous, binary, unordered categorical and ordered categorical data. In addition, MICE can impute continuous two-level data, and maintain consistency between imputations by means of passive imputation. Many diagnostic plots are implemented to inspect the quality of the imputations.

For this study, a CART method is used for imputation using mice. The 1st iteration is chosen to combine with the original data.

Code:
```
#Impute data using MICE package from R, 1st iteration is used
imputedData<- mice(hloan,m=2,maxit=5,meth='cart',seed=500)
hloan <- complete(imputedData,1)
```

- Fill by mean/median, or mode

Null values in continuous variables:

- may be filled or replaced with either mean or median values depending on the existence of outliers. If outliers are present, using average mean to replace nulls will be useless as it is heavily affected by outliers, thus median will be used instead.

Code:
```
#Impute data using Median value of oontinuous variables
Ex. hloan$LoanAmount[is.na(hloan$LoanAmount)]
<-median(hloan$LoanAmount,na.rm=TRUE)
```

Null values in categorical variables:

- may be replaced by the most occurring value in their respective column.

Code:
```
#Impute data using most frequent from categorical variables
Ex. hloan$Gender[is.na(hloan$Gender)]<- "Male"
```

iii.a.1.2 Treatment of Outliers

By producing kernel density plot, and box plot of variables, we can observe the existence of many outliers, causing the skewness of the plots. Outlier visualization is under Data Visualization Figures 5-9.

To eliminate the bias of wide ranges caused by outliers and the differences in variable units to the predicting model later on, data may be log transformed, standardized or normalized.The choice may also differ depending on the machine learning algorithm requirement.A preprocessed, and transformed data may or may not cause a boost in accuracy.

Table 4 Treatment of Outliers

1. Remove outliers 2. Standardize continuous data Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. Formula: $X' = \frac{X - \mu}{\sigma}$ 3. Normalize continuous data Normalization using z-transformation formula zVar <- (myVar - mean(myVar)) / sd(myVar) 4. Log Transform

iii.a.2 Feature Engineering and Scaling

- Investigate relationships between variables
- Consider adding new variables based on established relationships between variables. This is only doable by having knowledge regarding the domain so it is of utmost importance to research and identify other approaches of analysis. Some variables that may be added are the following:

    - **Home Loan EMI or Equated Monthly Installment (EMI)**[2]

      An equated monthly installment (EMI) is a fixed payment amount made by a borrower to a lender at a specified date each calendar month.

      Home loan EMI is the amount that is paid to the lender for the purpose of repayment of the borrowed loan to finance your home. At the time of availing a home loan, EMI is calculated by your lending institution based on the borrowed amount, sanctioned rate of interest and loan tenure.

[2]The mathematical formula to calculate EMI is:

$$A = P \times \frac{r\,(1+r)^{n}}{(1+r)^{n} - 1}$$

Where:

A = Periodic EMI amount

P = Principal borrowed

r = Periodic interest rate (annual interest rate/12)

n = Total number of payment (number of months during the loan tenure)

For this study: r = 0.01167; for annual interest of 14%

- **Total Income**

  Sum of Applicant Income and Coapplicant Income

- **Debt to Income Ratio**

  A debt-to-income ratio (DTI) is a personal finance measure that compares the amount of debt you have to your overall income. Lenders, including issuers of mortgages, use it as a way to measure your ability to manage the payments you make each month and repay the money you have borrowed.

  Lenders prefer to see a debt-to-income ratio smaller than 36%, with no more than 28% of that debt going towards servicing a home loan

Code: Create Functions and add new columns for new variables

```
#Create functions for new features
emi <- function(p, r, n){
  return((p*r*(1+r)**n)/(((1+r)**n)-1))
}

debt_to_income_ratio <- function(ti,emi){
  return((emi)/ti)
}
```

[2] 1 https://corporatefinanceinstitute.com/resources/knowledge/credit/equated-monthly-installment-emi

```
#Use functions to create new features (Total Income, EMI, Debt
to Income Ratio)
hloan$TotalIncome <-
hloan$ApplicantIncome+hloan$CoapplicantIncome
hloan$EMI <- emi(p=(hloan$LoanAmount*1000), r=0.01167,
n=hloan$Loan_Amount_Term)
hloan$DebtToIncomeRatio <-
debt_to_income_ratio(hloan$TotalIncome, hloan$EMI)

#Log transform continuous variables
hloan$TotalIncome_log <- log(hloan$TotalIncome)
hloan$EMI_log <- log(hloan$EMI)
hloan$DebtToIncomeRatio_log <- log(hloan$DebtToIncomeRatio)
```

iii.b Visualization of Cleansed and Transformed Data (R, Power BI, Tableau, Python)

- Visualization of Distribution of Population
- create graphical plots per variable to determine imbalances in data distribution within the population and classification, or linear/nonlinear/polynomial relationships between independent variables, and create observations For example: Samples collected for Gender A are more than of Gender B.

iii.c Hypothesis Formulation

- Create hypothesis based on observations from produced visualizations

iii.d Hypothesis Testing by Statistics Analysis (R, Power BI, Python)

- Statistical Analysis (R,Power BIPython) - verify created hypotheses by hypothesis testing using parametric or non-parametric tests.
- To determine which method to use, it is important to understand first the requirements of each method.
- Rank variables according to relative importance to predicting Loan_Status by comparing p-values,covariance, correlation, and other statistical metrics.
- Hypothesis Testing in more detail will be explained in the Hypothesis Testing Chapter.

iii.e Modeling

- Using the results from hypothesis testing, create and improve predictive models.
- Compare models based on accuracy, precision, recall, f value, auc and other possible metrics to justify which is the best model. An

imbalanced dataset such as in this study will rely more on f value and roc curve.

- Select the best model to satisfy the specified objectives.

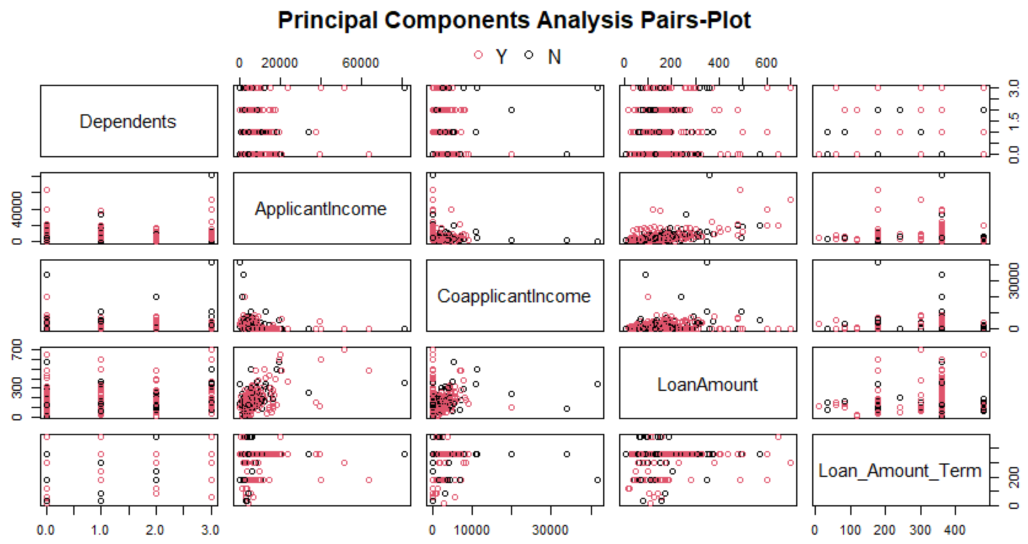
iii.f Model Improvement

- It can be done by boosting, tuning, feature selections or ensemble creations.

d. Data Visualizations

Figure 5 PCA Plot

Using pairs() function, we can produce a pair component analysis plot to visualize linear/nonlinear/polynomial relationships between variables. For raw visualization purposes, the non-transformed data is used.

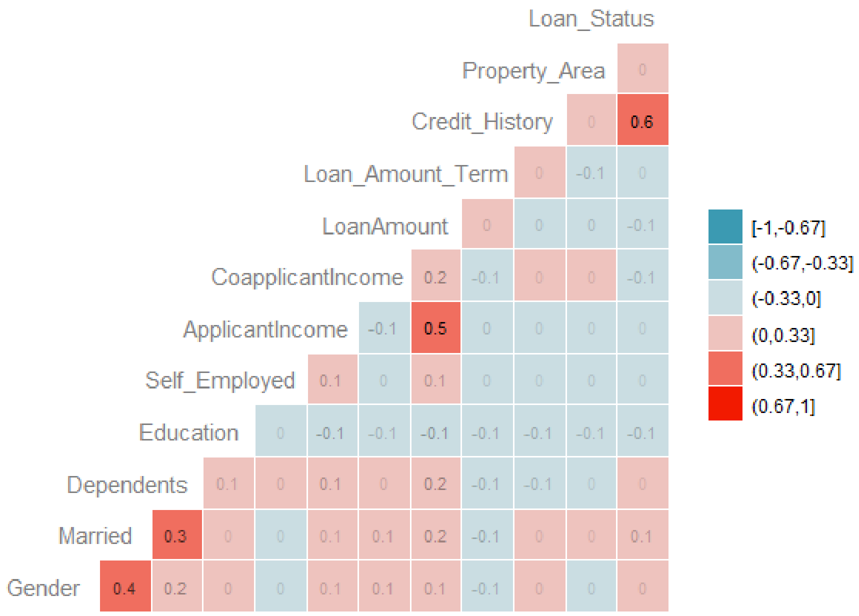


Code:

```
#Convert factors to numeric
numeric <- hloan %>% dplyr::select(where(is.numeric))
factor<- hloan %>% dplyr::select(where(is.factor))
#Create pairs Plot
pairs(factor[-7],     col=hloan$Loan_Status,main     =     "Principal
Components Analysis Pairs-Plot")
```

A correlation matrix is also produced to supplement the observation from PCA pairs-plot.

Figure 6 Correlation Matrix



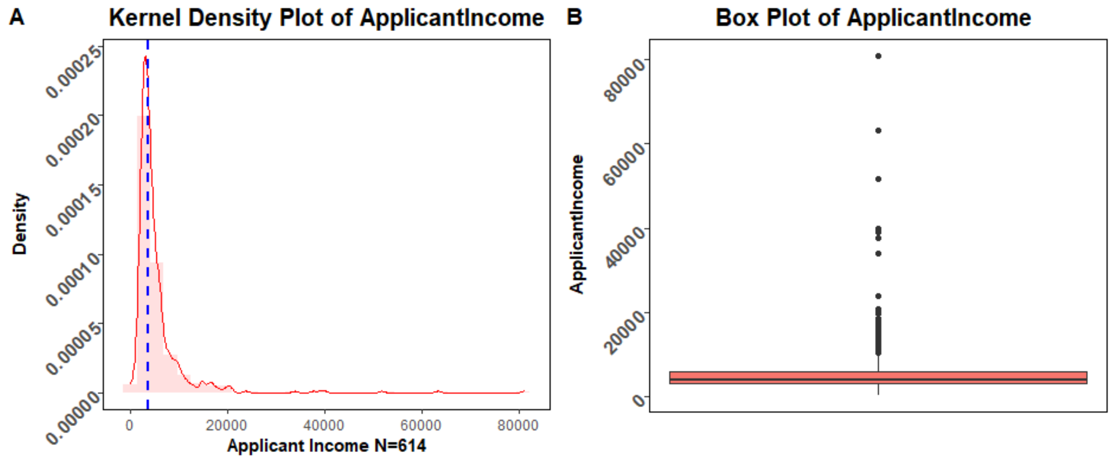
Code:

```
#Load GGally library
library(GGally)
# Convert data to numeric
corr <- data.frame(lapply(hloan[-1], as.numeric))
corr
# Plot the graph
ggcorr(corr,
       method = c("pairwise", "pearson"),
       nbreaks = 6,
       hjust = 0.9,
       label = TRUE,
       label_size = 3,
       label_alpha = TRUE,
       color = "grey50",
       layout.exp = 1)
```

Observation: There may be strong correlation in Credit History and Loan Status. Which will be further explained in the hypothesis and testing section.

Table 5 Visualization and Observation of Distribution of Population

Figure 7 Applicant Income



Observation: With possible Outliers
Form: Gaussian-like distribution, but positively skewed
Median(x1000): P3812.5
Reasons for outliers in the Income related variables may be the economic differences of status quo of applicants.

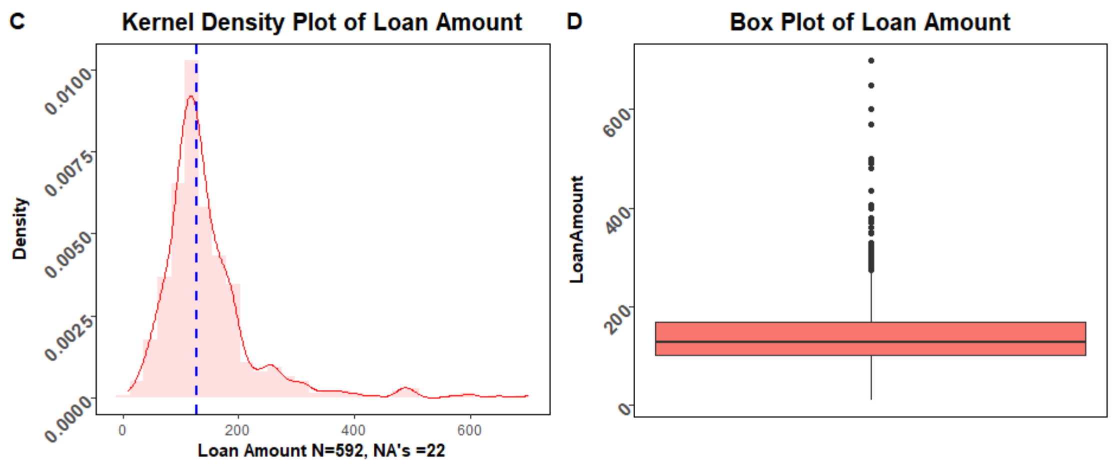
It can also easily be observed by summarizing the continuous variables in R, and comparing the ranges (Max-Min) with respect to their median.

Code:

```
#Load ggplot2 library, and ggpubr
library(ggplot2)
library(ggpubr)
#Create histograms/density plots for continuous variables
density_ai <- ggplot(hloan, aes(ApplicantIncome)) +
  xlab("Applicant Income N=614") + ylab("Density")+
  geom_histogram(alpha = .2, fill = "#FF6666",aes(y =
stat(density))) +mytheme+
  geom_density(col = "red")+
  ggtitle("Kernel Density Plot of ApplicantIncome")+mytheme+
  geom_vline(aes(xintercept=median(ApplicantIncome)),
             color="blue", linetype="dashed", size=1)
box_ai <- ggplot(hloan, aes(y = ApplicantIncome,
fill="ApplicantIncome")) +
  geom_boxplot()+mytheme +
  ggtitle("Box Plot of ApplicantIncome ")+
    theme(axis.ticks.x = element_blank(),
          axis.title.x= element_text(size=22,face="bold"),
          legend.title = element_blank(),
          legend.position = "none")+
  xlab("")
#Arrange Plots
ggarrange(density_ai, box_ai + rremove("x.text"),
```

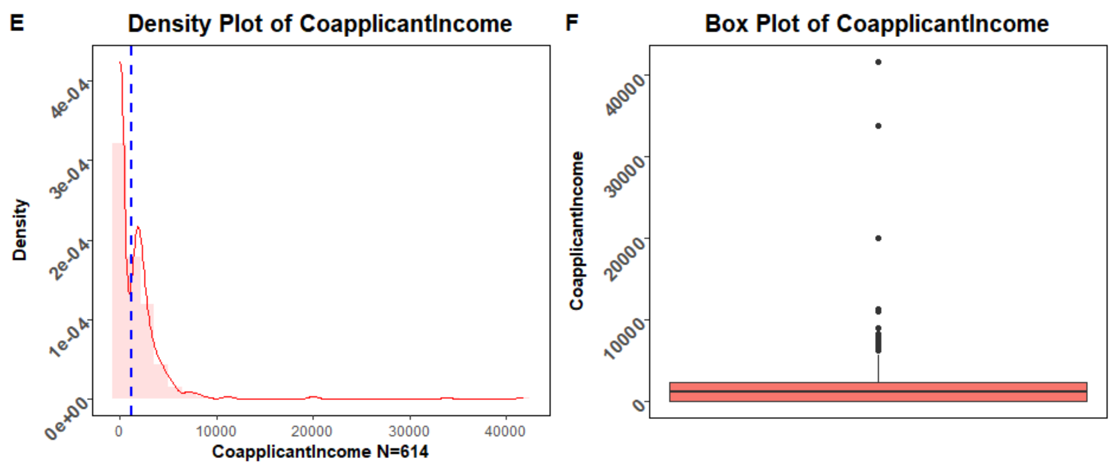
```
        labels = c("A", "B"),
                ncol = 2,
                widths = c(2.7, 2.7), heights = c(0.2, 2.5))
```

Figure 8 Loan Amount



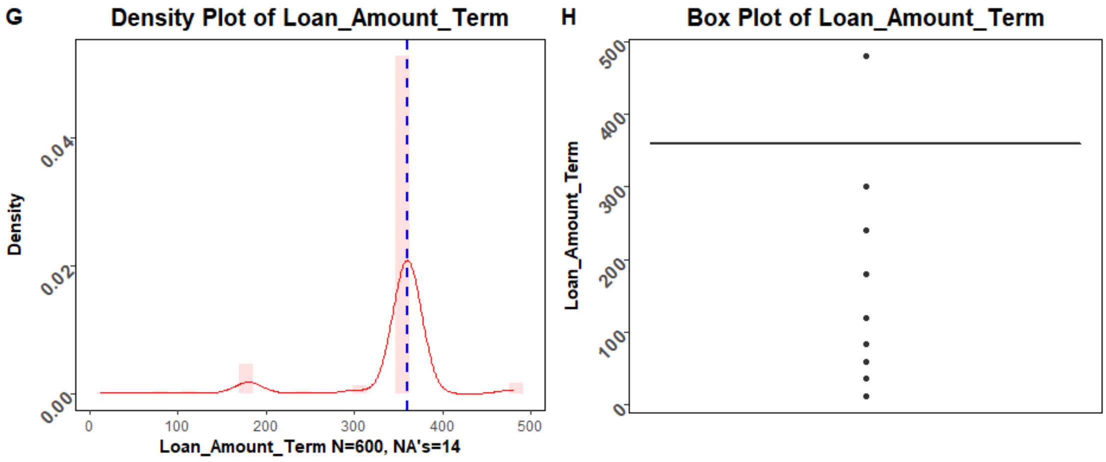
Observation: Loan Amount showing possible outliers
Form: Gaussian-like distribution, but positively skewed
Median(x1000):P128

Figure 9 Coapplicant Income



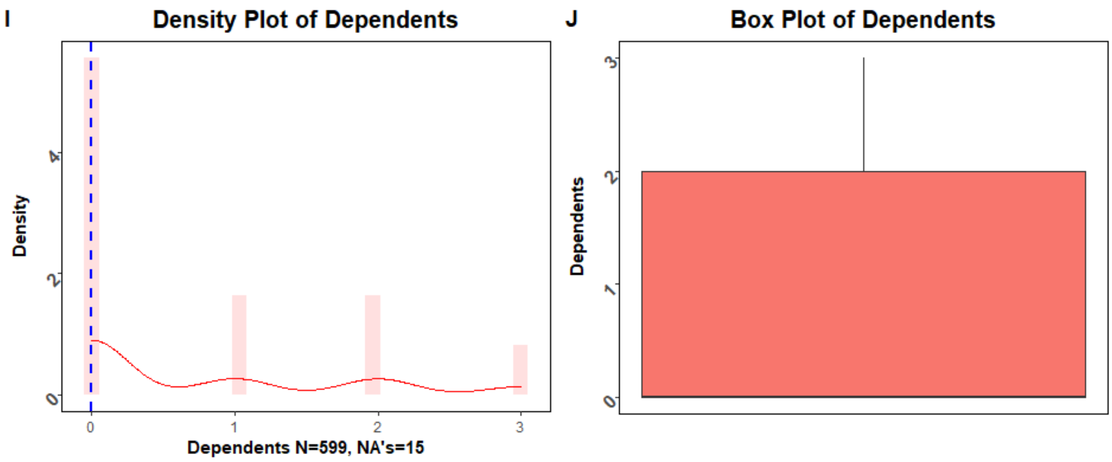
Observation: showing possible outliers
Form: Non-gaussian like distribution, positively skewed
Median(x1000):P1188.5

Figure 10 Loan Amount Term (in months)



Observation: Non-Gaussian like distribution, left skewed
Median (months):360

Figure 11 Dependents



Observation: Non-Gaussian like distribution
Median: 0

Table 6 Outlier Treatments

Type of Distribution	Variable	Observation	Type of Transformation
Resembles a Gaussian(normal) distribution:	ApplicantIncome, CoapplicantIncome, LoanAmount	same scale,	Standardization or log transform

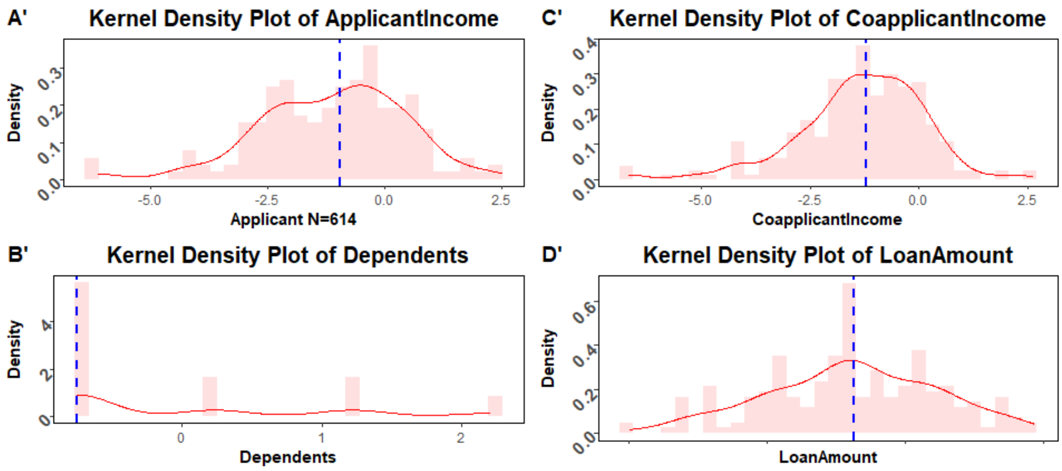
Resembles a non-Gaussian distribution:	Loan_Amount_Term, Dependents	High concentration of values in only a few scattered levels.	Normalization

In some case studies observed, log transformed data performs better in creating predictive models, as it eliminates the skewness of data distribution, which is important to take into consideration as hypothesis/statistical testing methods and model algorithms assume that data being tested is normally distributed.

In this study however, fitting the model to raw, normalized and standardized data may be done to compare the performance for best results.

Table 7 Visualization of continuous variables after log transformation

Figure 12



Plots are less skewed and has clearer resemblance to a normal distribution

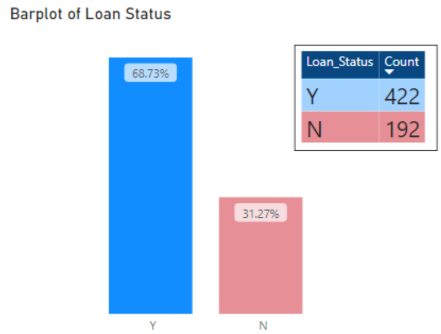
Code:

```
#plots after log transformation
#Log Transformed Applicant Income Plot
sc_ai <- ggplot(hloan, aes(ApplicantIncome)) +
  xlab("Applicant N=614") + ylab("Density")+
  geom_histogram(alpha = .2, fill = "#FF6666",aes(y =
stat(density))) +mytheme+
  geom_density(col = "red")+
  ggtitle("Kernel Density Plot of ApplicantIncome")+mytheme+
  geom_vline(aes(xintercept=median(ApplicantIncome, na.rm =
TRUE)), color="blue", linetype="dashed", size=1)+
  scale_x_log10()
#Log Transformed  Loan Amount plot
sc_la <- ggplot(hloan, aes(LoanAmount)) +
```

```
xlab("LoanAmount") + ylab("Density")+
  geom_histogram(alpha = .2, fill = "#FF6666",aes(y =
stat(density))) +mytheme+
  geom_density(col = "red")+
  ggtitle("Kernel Density Plot of LoanAmount")+mytheme+
  geom_vline(aes(xintercept=median(LoanAmount, na.rm = TRUE)),
             color="blue", linetype="dashed", size=1)+
  scale_x_log10()
#Log Transformed Dependents plot
sc_d <- ggplot(hloan, aes(Dependents)) +
  xlab("Dependents") + ylab("Density")+
  geom_histogram(alpha = .2, fill = "#FF6666",aes(y =
stat(density))) +mytheme+
  geom_density(col = "red")+
  ggtitle("Kernel Density Plot of Dependents")+mytheme+
  geom_vline(aes(xintercept=median(Dependents, na.rm = TRUE)),
             color="blue", linetype="dashed", size=1)+
  scale_x_log10()
#Log Transformed Coapplicant Income plot
sc_cai <- ggplot(hloan, aes(CoapplicantIncome)) +
  xlab("CoapplicantIncome") + ylab("Density")+
  geom_histogram(alpha = .2, fill = "#FF6666",aes(y =
stat(density))) +mytheme+
  geom_density(col = "red")+
  ggtitle("Kernel Density Plot of CoapplicantIncome")+mytheme+
  geom_vline(aes(xintercept=median(CoapplicantIncome, na.rm =
TRUE)),
             color="blue", linetype="dashed", size=1)+
  scale_x_log10()
#Arrange the plots
ggarrange(sc_ai, sc_cai, sc_d, sc_la + rremove("x.text"),
          labels = c("A'", "C'", "B'", "D'"),
          ncol = 2,
          nrow = 2)
```

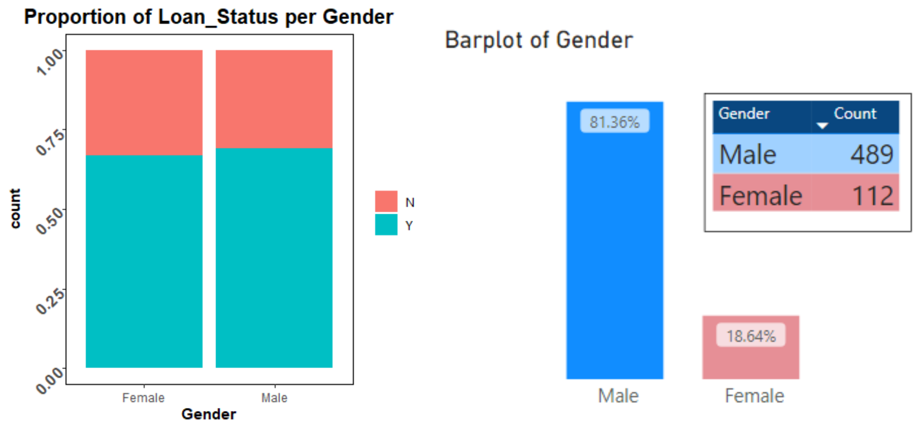
To better illustrate the population differences per variable, Power BI and R are utilized. This will be useful later for statistical proportional comparison by average to determine if a higher sample has any effect on the probability of getting a Y on Loan_Status.

Figure 13 Loan Status (Response Variable)



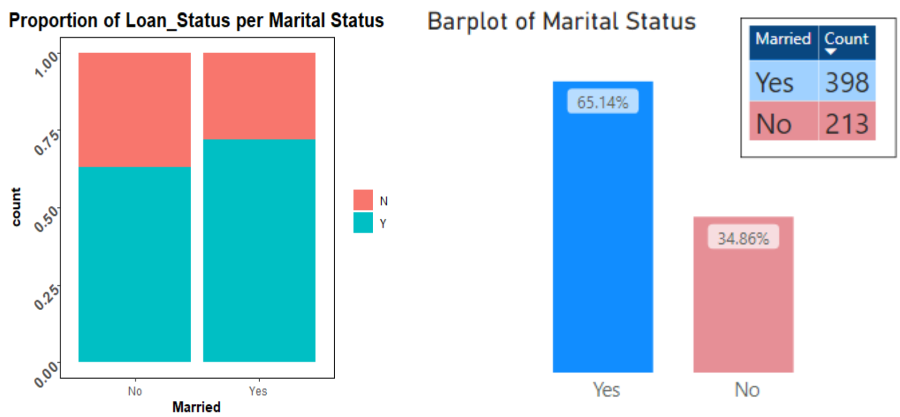
Observation: More applicants are eligible for a house loan. This denotes imbalance in classification and should be taken into consideration when building machine learning models

Figure 14 Gender



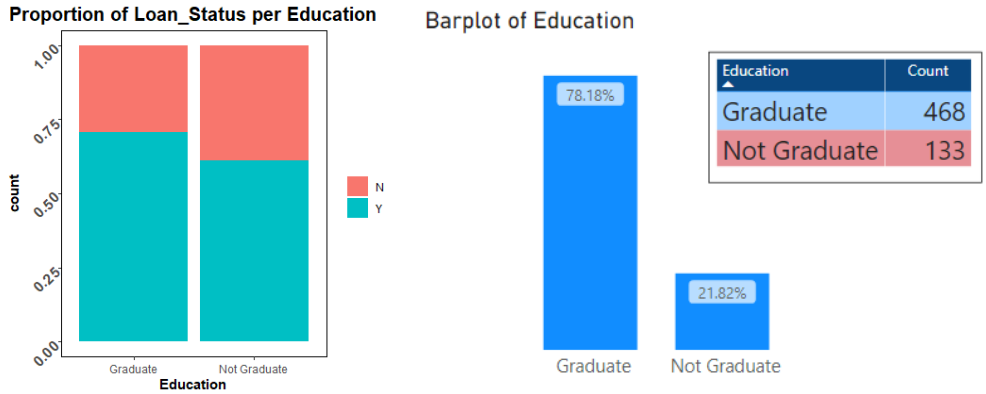
Observation: There are more males in the dataset. However the proportions of Loan_Status results with respect to both genders look almost equal.

Figure 15 Marital Status



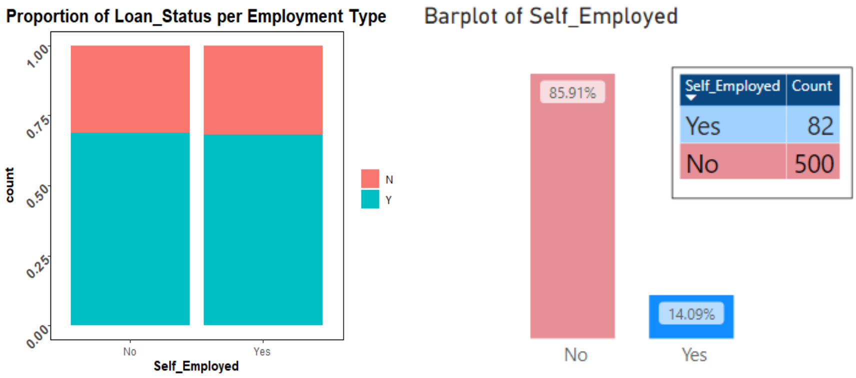
Observation: There are more married applicants in the dataset. The proportion of Y on Loan_Status is greater for the married applicants.

Figure 16 Education



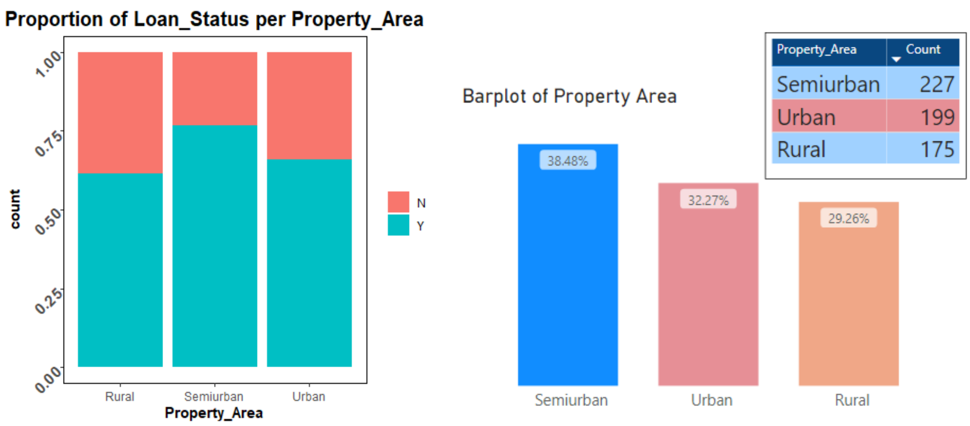
Observation: There are more graduates in the dataset. Graduates also have a higher proportion of Y on Loan_Status.

Figure 17 Employment



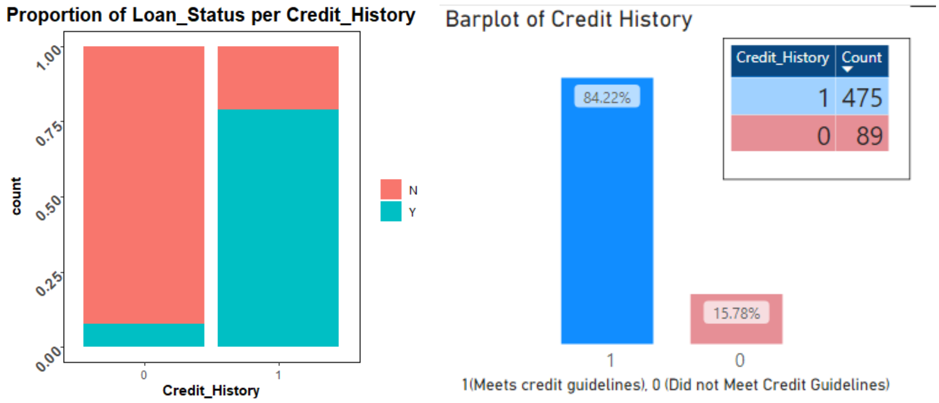
Observation: There are more self-employed in the dataset. The proportions of Loan_Status results with respect to employment categories look the same.

Figure 18 Property Area



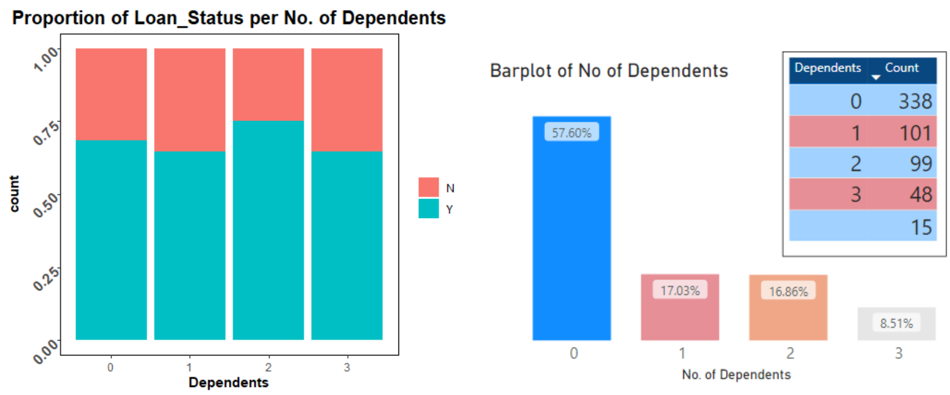
Observation: More people who live in the Semiurban area are part of the dataset. Semiurban area also has the highest proportion of Y on Loan Status.

Figure 19 Credit History



Observation: More applicants who met the credit history guidelines are part of the dataset. The proportion of Y on Loan_Status is higher when Credit_History is valued 1.

Figure 20 Dependents

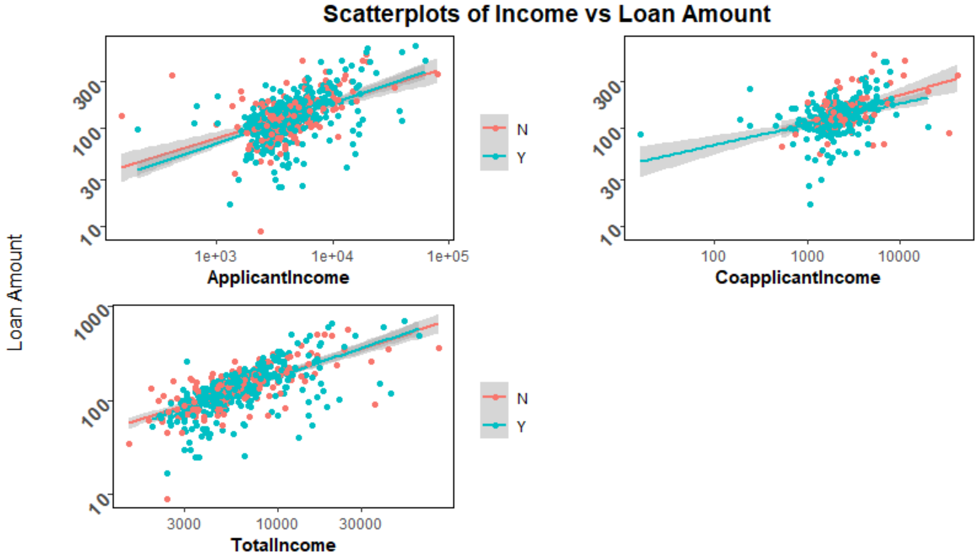


Observation: There are more applicants who have zero(0) dependents. The proportions of Y on Loan_Status do not greatly vary between different numbers of dependents.

Code:

```
#Plot a bar graph for Dependents
ggplot(hloan, aes(x = Dependents, fill = Loan_Status)) +
  geom_bar(position = "fill") +
  mytheme +
  ggtitle("Proportion of Loan_Status per No. of
Dependents")
theme(axis.text.x = element_text(angle = 90))+
  coord_cartesian(clip = 'off')
```

Figure 21 Income vs Loan Amount



ALPHA FINANCING dataset

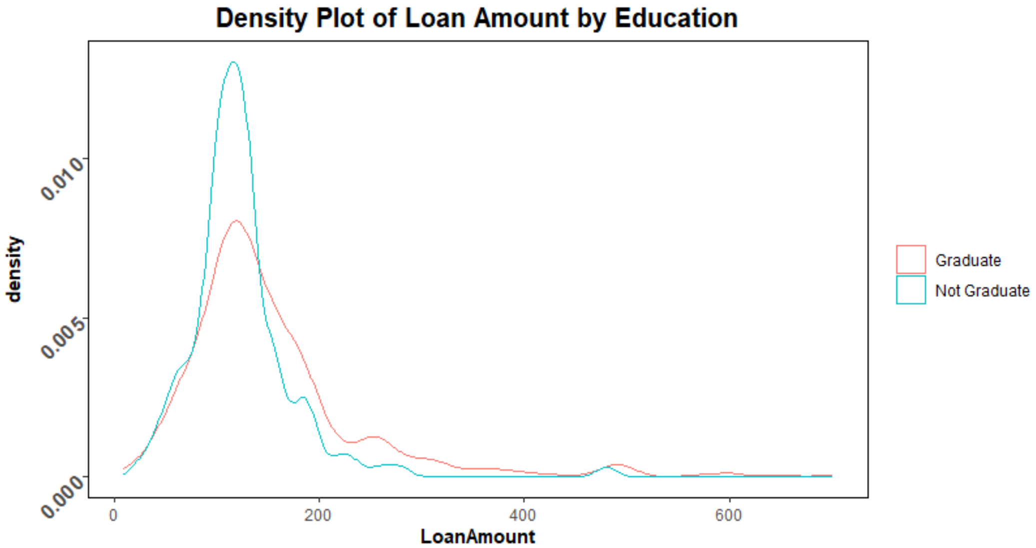
Observation: There is a linear relationship between Income and Loan Amount.

Code:

```
# Scatterplot of ApplicantIncome and LoanAmount: log
transformed
ai_la <- ggplot(hloan, aes(ApplicantIncome,
LoanAmount,col=hloan$Loan_Status)) +
geom_smooth(method = "lm") +
geom_jitter()+
ylab("")+
scale_x_log10() + scale_y_log10()+mytheme

# Scatterplot of CoapplicantIncome and CoapplicantIncome

cai_la <- ggplot(hloan, aes(CoapplicantIncome, LoanAmount,
col=hloan$Loan_Status)) +
geom_smooth(method = "lm") +
geom_jitter()+
ylab("")+
scale_x_log10() + scale_y_log10()+mytheme

# Scatterplot of TotalIncome and LoanAmount
ta_la <- ggplot(hloan, aes(TotalIncome,
LoanAmount,col=hloan$Loan_Status)) +
geom_smooth(method = "lm") +
geom_jitter()+
ylab("")+
scale_x_log10() + scale_y_log10()+mytheme
#Arrange the plots
figure1 <- ggarrange(ai_la, cai_la, ta_la,
ncol = 2,
nrow=2)
annotate_figure(figure1,
top = text_grob("Scatterplots of Income vs Loan Amount",
```

```
color = "black", face = "bold", size = 14),
bottom = text_grob("ALPHA FINANCING dataset", color =
"blue",
hjust = 1, x = 1, face = "italic", size = 10),
left = text_grob("Loan Amount", color = "black", rot = 90),
right = text_grob(bquote(""**""), rot = 90),
fig.lab = "", fig.lab.face = "bold")
#Divide the console window into 2 row and 2 columns
par(mfrow=c(2,2))
```

Figure 22 Loan Amount by Education

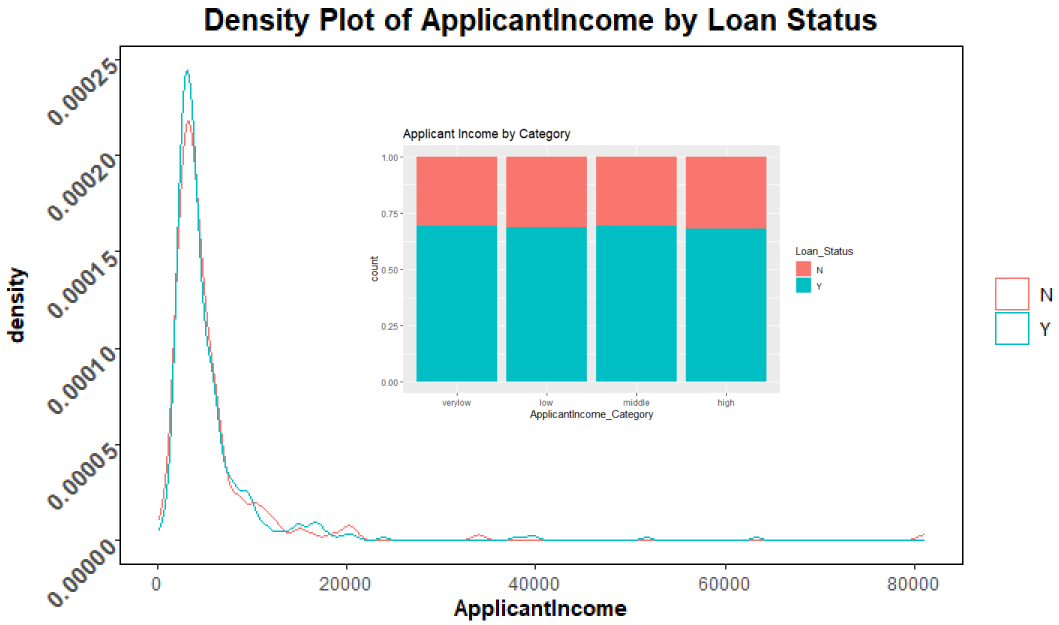


Observation: There is a non-homogeneous distribution of Loan amount by Graduates and non-graduates.

Code:

```
# You can check the density of the Loan Amount by type of
education.
ggplot(hloan, aes(x = LoanAmount)) +
geom_density(aes(color = Education), alpha = 0.5) +
mytheme+
 ggtitle("Density Plot of Loan Amount by Education")
```

Figure 23 Loan Status by ApplicantIncome

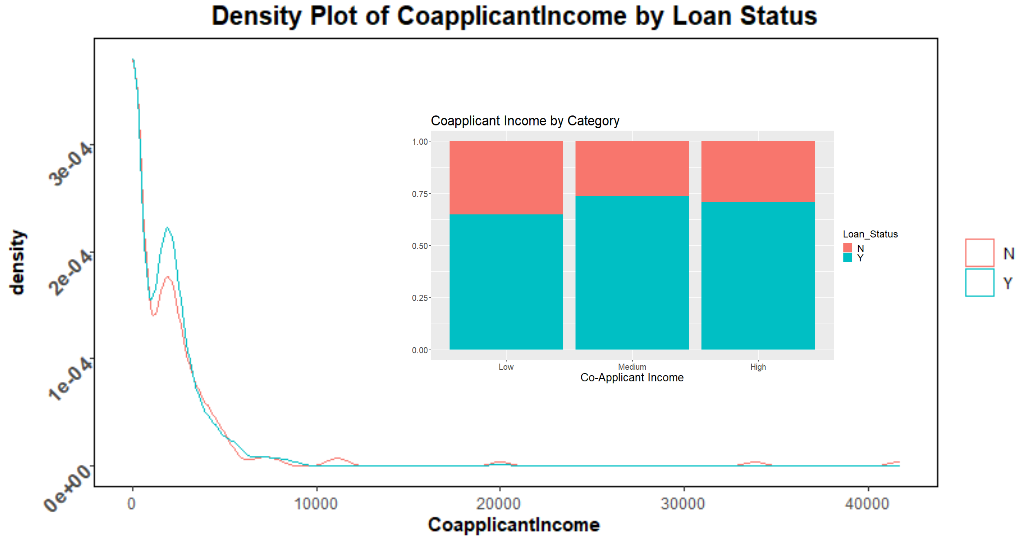


Observation: Proportion of Y on Loan status appears to be the same regardless of Applicant Income category.

Code: FEATURE ENGINEERING

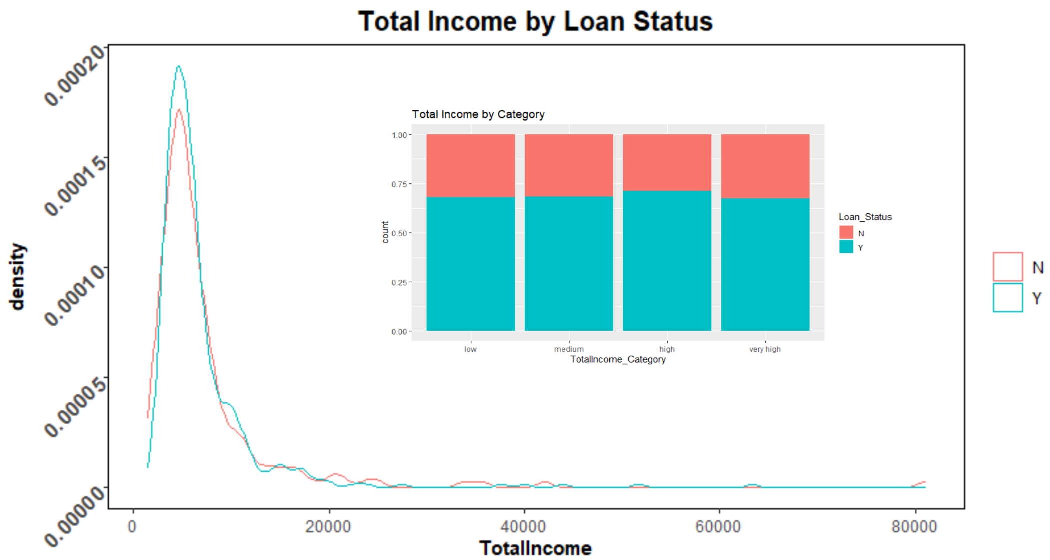
```
#Cut Continuous variables into categories
ai <- hloan %>%
mutate(ApplicantIncome_Category=cut(ApplicantIncome,
breaks=unique((quantile(ApplicantIncome,
probs=seq.int(0,1, by=1/4)))),
labels=c("verylow","low","middle","high")))

#plot of categorized continuous variables
filter(ai, !is.na(ApplicantIncome_Category)) %>%
  ggplot() +
  geom_bar(mapping = aes(x = ApplicantIncome_Category,
fill = Loan_Status), position = "fill") +
  scale_fill_discrete(na.translate=FALSE)+
  ggtitle("Applicant Income by Category")
```

Figure 24 Loan Status by CoapplicantIncome



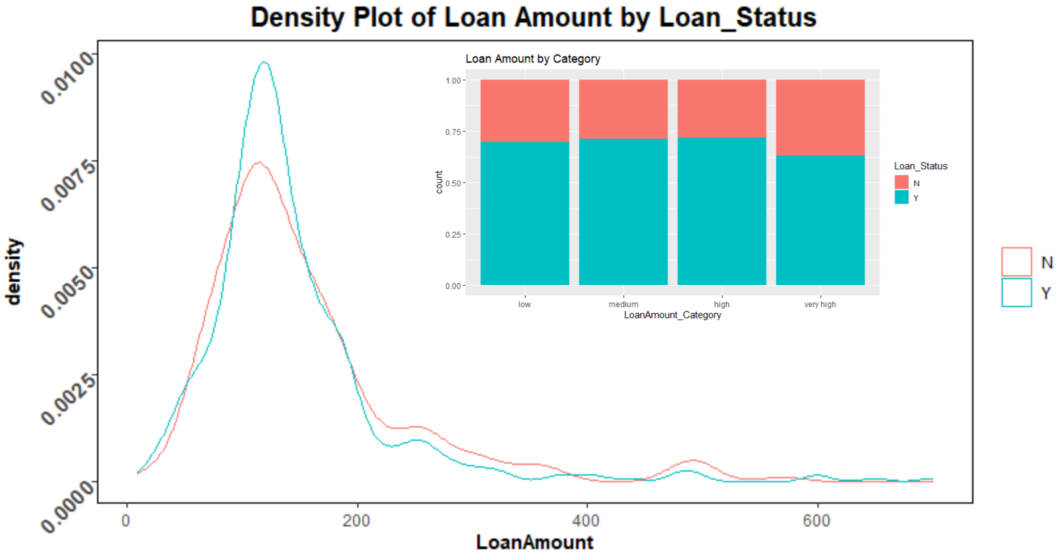
Observation: Proportion of Y on Loan status is highest in the low income category.

Figure 25 Loan Status by Total Income



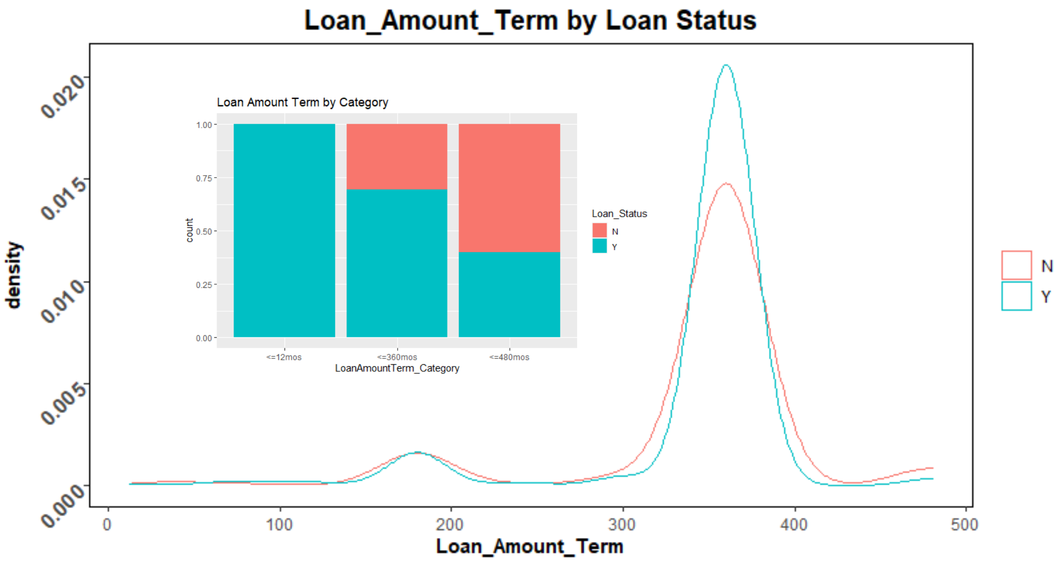
Observation: Proportion of Y on Loan status appears to be the same regardless of Income category.

Figure 26 Loan Status by Loan Amount



Observation: Proportion of Y on Loan status appears to be the same regardless of Loan Amount category.

Figure 27 Loan Status by Loan Amount Term (in months)



Observation: Proportion of Y on Loan status is highest on Loan Terms below or equal to 12months.

Summary of Observations:

In the data visualization process, we can observe some issues. These are the missing data, outliers, and data imbalance. While some categories may be dominating in number in their respective fields (eg Male, Self_Employed, Graduates, 0(zero) dependents applicants), it is noticeable how such dominance may not have the same effect on the proportion of Y and N on Loan Status. This is a good subject for hypothesis, to confirm if the number of observations per category directly affects the outcome of Loan_Status.

Other variables that are showing more consistent evidence of relationship with Loan_Status in terms of number of observations and proportion of Y on Loan_Status are Credit_History, Education, Property_Area, and Marriage.

Applicant Income, Coapplicant Income, Total Income, Loan Amount, Loan_Amount_Term, and number of Dependents show either unclear or no variations with respect to Loan_Status.